

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Claims 1-5 (Cancelled):

Claim 6 (Currently Amended): A method of accessing a single resource in an operating system, comprising the steps of:

- incrementing a counter included in said operating system by a discrete value, upon generation of a resource, and using a counter value as generation identifying information;
- assigning to said resource an identifier composed of address information and the generation identifying information of said resource;
- storing said generation identifying information at a leading location of said resource;
- extracting generation identifying information from an identifier transferred as an argument of a system call issued by one user application for accessing said resource;
- comparing the extracted generation identifying information with the generation identifying information stored in said resource at said leading location; and
- enabling access to said resource when coincidence is found between both of said generation identifying information, while disabling access to said resource when discrepancy is found between both of said generation identifying information.

wherein a given one of a plurality of processes in said operating system using said resource is executed by:

acquiring said resource for use by a given one process after
disabling abortion and preemption of said given one process;
clearing said given one process from a preempt-disabled state
and disabling preemption of said given one process after processing
for said resource;

clearing said given one process from the preempt-disabled state
and from an abort-disabled state after said resource has been
deallocated from use by said given one process; and

executing a forcive termination request issued for said given one
process during a period in which said given one process has been in
the abort-disabled state.

Claim 7 (Previously Presented): An accessing method according to claim 6, wherein said generation identifying information corresponds to a time said resource was generated.

Claim 8 (Previously Presented): An accessing method according to claim 6, wherein said identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said resource and a 32 least significant bits which indicates the generation identifying information.

Claim 9 (Previously Presented): An accessing method according to claim 6, further comprising:

when said coincidence is found between both the generation identifying information, determining that said identifier of said resource is valid and enabling access to said resource using the resource address to perform the processing of said resource; and

when said discrepancy is found between both the generation identifying information, determining that said identifier of said resource is invalid and disabling access to said resource.

Claim 10 (Previously Presented): An accessing method according to claim 9, wherein said identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said resource and a 32 least significant bits which indicates the generation identifying information.

Claim 11 (Previously Presented): An accessing method according to claim 10, wherein said generation identifying information contains leading 16 bits representing said count value of said counter and trailing 16 bits representing a process identifier.

Claim 12 (Previously Presented): An accessing method according to claim 9, wherein said counter records the number of times said resource is generated, and is set to an initial value of zero and incremented by one each time said resource is generated.

Claim 13 (Currently Amended): A method for controlling access to a shared resource in an operating system from different processes, comprising the steps of:

incrementing a counter by a discrete value each time a shared resource is requested from a user application for recording a number of times said shared resource is requested from said user application;

assigning to said shared resource requested, a resource identifier comprised of a resource address and generation identifying information assigned to said shared resource;

storing said generation identifying information of said resource identifier at a designated location of said shared resource;

extracting generation identifying information from a resource identifier transferred as an argument of a system call issued by said user application request for accessing said shared resource;

making a comparison between the extracted generation identifying information with the generation identifying information stored at said designated location of said shared resource; and

controlling access to said shared resource in response to said comparison, wherein a given one of different processes in said operating system using said shared resource is executed by:

acquiring said shared resource for use by a given one process
after disabling abortion and preemption of said given one process;

clearing said given one process from a preempt-disabled state
and disabling preemption of said given one process after processing
for said shared resource;

clearing said given one process from the preempt-disabled state
and from an abort-disabled state after said shared resource has been
deallocated from use by said given one process; and

executing a forcive termination request issued for said given one
process during a period in which said given one process has been in
the abort-disabled state.

Claim 14 (Previously Presented): A method according to claim 13,
further comprising:

enabling access to said shared resource when the extracted generation
identifying information matches the generation identifying information stored at said
designated location of said shared resource; and

disabling access to said shared resource when the extracted generation
identifying information does not match the generation identifying information stored
at said designated location of said shared resource.

Claim 15 (Previously Presented): A method according to claim 14,
wherein said shared identifier corresponds to a 64-bit identifier containing a 32 most
significant bits which indicates the address of said shared resource and a 32 least
significant bits which indicates the generation identifying information.

Claim 16 (Previously Presented): A method according to claim 14, wherein said generation identifying information contains leading 16 bits representing a count value of said counter and trailing 16 bits representing a process identifier.

Claim 17 (Previously Presented): A method according to claim 13, further comprising:

when the extracted generation identifying information matches the generation identifying information stored at said designated location of said shared resource, determining that said resource identifier of said shared resource is valid and enabling access to said shared resource using the resource address to perform the processing of said shared resource; and

when the extracted generation identifying information does not match the generation identifying information stored at said designated location of said shared resource, determining that said resource identifier of said shared resource is invalid and disabling access to said shared resource.

Claim 18 (Previously Presented): A method according to claim 17, wherein said resource identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said shared resource and a 32 least significant bits which indicates the generation identifying information.

Claim 19 (Previously Presented): A method according to claim 18, wherein said generation identifying information contains leading 16 bits representing a count value of said counter and trailing 16 bits representing a process identifier.

Claim 20 (Previously Presented): A method according to claim 13, wherein said counter records the number of times said shared resource is requested, and is set to an initial value of zero and incremented by one each time said shared resource is requested.

Claim 21 (Previously Presented): A method according to claim 19, wherein said counter records the number of times said shared resource is requested, and is set to an initial value of zero and incremented by one each time said shared resource is requested.

Claim 22 (Canceled):

Claim 23 (Currently Amended): A method according to claim 22, further comprising:

providing a queue for use of said resource for registering those processes issued respective requests;

executing in a multiprocessing environment a leading one of the processes registered in said queue and issued respective requests for use of said resource; and

driving periodically processing relating to said process after completion of execution and executing serially the processes registered in said queue.

Claim 24 (Canceled):

Claim 25 (Currently Amended): A method according to claim 24¹³,
further comprising:
providing a queue for use of said shared resource for registering those
processes issued respective requests;
executing in a multiprocessing environment a leading one of the processes
registered in said queue and issued respective requests for use of said shared
resource; and
driving periodically processing relating to said process after completion of
execution and executing serially the processes registered in said queue.